White Paper Report

Report ID: 98510

Application Number: HD5078209

Project Director: Peter Shillingsburg (pshillingsburg@luc.edu)

Institution: Loyola University, Chicago

Reporting Period: 9/1/2009-1/31/2011

Report Due: 4/30/2011

Date Submitted: 4/1/2011

hrit

Final Performance Report:

Humanities Research Infrastructure and Tools

White Paper to the NEH Office of Digital Humanities

Level II Digital Humanities Start-Up Grant
HD-50782-09
1/09/2009 to 1/31/2011

Peter Shillingsburg
Loyola University Chicago
Department of English
1032 W. Sheridan Rd
Chicago, IL 60660
pshillingsburg@luc.edu

George Thiruvathukal
Loyola University Chicago
Department of Computer Science
820 N. Michigan Ave.
Chicago, IL 60611
thiruvathukal@gmail.com

Steve Jones
Loyola University Chicago
Department of English
1032 W. Sheridan Rd
Chicago, IL 60660
s3jones1@gmail.com

Abstract

HRIT (pron. *writ)* is a web environment of applications designed for humanists who are building, viewing, using, and enhancing scholarly textual projects. Starting with digital images of primary materials (the original texts) and with transcriptions, the textual project builder can use the environment and tools provided by HRIT to create customized views and interactive content from those raw materials. Textual variation is handled by the CorTex system. Commentary is handled by a standoff markup approach (supporting TEI and other markup schemes) with a variety of routines for creating commentary and for processing legacy files with embedded markup. CaTT (Collaborative Tagging Tool) is the basic standoff markup tool. For the user or viewer the E-Carrel provides easy access to images of individual copies of a work and their precisely identified transcriptions, annotations, commentary and links in a single workspace, where the user can customize the display of textual and scholarly resources. It further offers viewers the opportunity to contribute to the project through the use of tools and a vetting system customizable to the individual project. Collaboration and data integrity, which are traditionally incompatible concepts, are both supported, not only during initial project construction by a select collaborating team but also, subsequent to launch, by enabling continuing and broad on-line collaboration over time.

Although focused in the first instance on literary texts, the HRIT environment and capabilities are designed to be adaptable not only to texts in other fields, such as law, history,

medicine, music, and dance, but to fields whose primary materials are non-textual: images, three-dimensional objects, audio files, and video.

This report is also available on-line at https://sites.google.com/a/ctsdh.luc.edu/hrit-intranet/home?pli=1

1. Introduction

Textual scholars, creating scholarly editions whether in print or electronically, are generally resigned to the fact that very few readers will avail themselves of the full complexities of textual studies, historical explanations, critical commentaries, or extensive intra- and extra-mural linking to relevant sources. However, they have for centuries continued to develop and improve the mechanisms for scholarly editions to be comprehensive in collecting, analyzing, and presenting texts and contexts.

The HRIT system of environment and tools addresses the needs of both humanist scholars and viewers. For humanities scholars it attempts to advance the ability of textual scholars to provide comprehensive, accurate, reliable foundations for the historical and critical study of literary texts, and to provide for long-term continued collaborative critical and scholarly "superstructuring" of the electronic scholarly edition as a "knowledge site."[1]

For viewers, from serious scholars to those who simply want a text and selected kinds of information about it, HRIT endeavors to provide user-friendly access to original materials and commentary about them. To that end, we are designing a navigation system and are exploring ways to provide social networking and mobile device access to HRIT projects, data, and services.

HRIT grew out of the idea of an E-Carrel drafted in 2004, further developed in 2006 in Shillingsburg's *From Gutenberg to Google*, chapters 3 and 4, and combined with ideas for user-friendly tagging tools by Nicholas Hayward, for stand-off markup in several publications going back to 2001 by Paul Eggert, Graham Barwell, Chris Tiffin, and Phillip Berrie, and for merging text files and for detection and display of textual variants by Desmond Schmidt. These independent ideas and projects came together at the Loyola University Chicago Center for Textual Studies and Digital Humanities in 2008, where the HRIT team was formed to develop the HRIT environment.

The core idea of HRIT is that electronic scholarly textual projects consist of three separate categories of material, which should be created and stored separately, but which should be aggregated and viewed in conjunction according to the desires of viewers, rather than in predetermined forms dictated by project builders. The three categories are **images** (stored as CorPix), **transcriptions** (stored as CorTex), and analytical **commentary** (stored as CorCode).

[1] "Knowledge sites" is a term used in Peter Shillingsburg's *From Gutenberg to Google: Electronic Representations of Literary Texts* (Cambridge University Press, 2006)--a work which formed the theoretical foundation for the HRIT system.

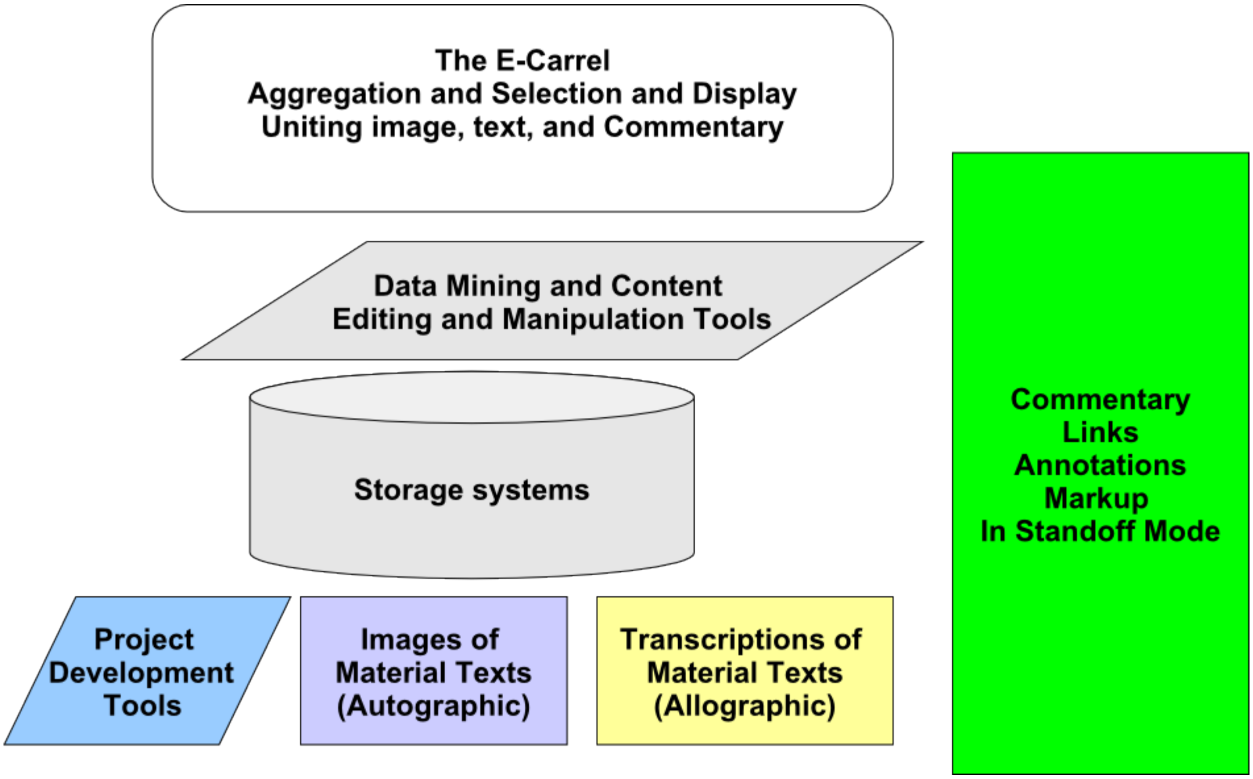

Fig. 1. The violet, yellow, and green rectangles represent the three categories of content for a textual project, respectively, the Iconic (image), the lexical (transcription), and the analytical (commentary) aspects of a scholarly project. The blue rhomboid represents the tools useful in creating a project's primary materials and placing them in the grey storage cylinder. The grey rhomboid represents the tools useful in extracting, unifying, and displaying the textual data from the three categories of content. The round-shouldered rectangle at the top represents the display or screen and the interface options available to the viewer.

A principle of component structure was chosen both for tool development and content structure and storage. The central aims were to give project **developers** three essential elements: 1) *protection* for the integrity of their work, 2) *credit* for their contributions to collaborative projects, and 3) endless *opportunity* for continued project expansion; and to give to **viewers** significant *choices* for selection and display of data mined from the project materials and an opportunity to join the originators in contributing new knowledge to the project.

In addition, HRIT has several design strategies to give electronic projects longevity:

- by making projects community owned and operated, not static immovable objects dependant on their originators for maintenance;
- by making the parts of any project separately maintainable through their component structures;
- by separating the foundation textual materials from the ever growing body of commentary, annotation, encoding, and markup, which is saved in stand-off, not embedded form; and
- by protecting projects from damage by multiplying and distributing copies of projects and their component parts on the Internet from which viewers, with E-Carrel, would aggregate project parts for local use.

This project is being built on work initially supported by the European Union through COST

Action 32, which supported a two-day workshop on CaTT in 2007 at De Montfort University, by the Office of Research at Loyola University Chicago, which provided an internal seed grant to the Center for Textual Studies and Digital Humanities, by the National Endowment for the Humanities through its level II digital humanities Start-Up Grants, and by the generous cooperation of uncompensated participants, in particular Federico Meschini, Desmond Schmidt, and Paul Eggert.

2. Participating Individuals

PI - Peter L. Shillingsburg, the Martin J. Svaglic Professor of Textual Studies, Loyola University Chicago, is former director of the Center for Textual Scholarship at De Montfort University, Leicester, UK, and author of three books on textual studies, two books on W. M. Thackeray, co-developer of PC-CASE (Computer Assisted Scholarly Editing) with Miriam Shillingsburg, Russell Kegley, and Boyd Nations, and general editor of the Michigan University Press edition of the Works of Thackeray.

Co-PI - George Thiruvathukal, Associate Professor of Computer Science, Loyola University Chicago, responsible for the quality of the computer programming and the technical interoperability amongst the modules of the project and between this project and other humanities projects. He has successfully produced computer projects for NSA and for corporations.

Co-PI - Steven Jones, Professor of English, Loyola University Chicago, responsible for the functional interoperability with other humanities tools projects and with data management solutions from the gaming world. He is Co-Creator and Co-Editor of Romantic Circles and a board member of NINES.

Senior Project Consultant - Paul Eggert, Australian Research Council Professorial Fellow, U of New South Wales at ADFA, Australia, is to consult on all aspects of the overall design of the project. As the project director for JITM, Eggert was the first humanities textual scholar and digital humanist to explore seriously the relevance of stand-off mark-up for collaborative projects and separation of foundation texts from critical enhancements.

Programmer - Dr. Nicholas Hayward, Honorary Senior Research Fellow, School of English, De Montfort University, UK. PhD in Ancient History and Archaeology and MSc in Computer Science. He is currently responsible for image development tools, tagging tools, TEI legacy tools, and the ‘mojulem’ framework for project publication and design. He is also the chief programmer and interface designer for the Woolf Online project, which is implementing HRIT tools and principles for the second phase of this ongoing project. He is also responsible for development and design of the Malory Project at Keio University, Japan, the Modernist Magazines and Elizabeth Jennings projects at De Montfort University, UK, and is a consultant to the Early Illustrated Books project at Keio University, Japan.

Programmer - Pedro Prieto Alarcon developed the services framework for HRIT and implemented the alpha version of the CorTex builder.

Consultatant and programmer - Dr. Desmond Schmidt, Queensland University of Technology, Brisbane, Australia, has PhDs in both Classics (University of Cambridge) and Information Technology (University of Queensland). Though unpaid, he readily associated with the HRIT team and adapted his MVD programs to the HRIT environment. He has also worked on HRIT API and stand-off markup mechanisms and participated in the HRIT day conference in Oct. 2010. He has added his NMerge and MVD programs to HRIT.

Consultant - Federico Meschini, unpaid, is the principle design liaison between the humanist and technical sides of this project. His extensive investigation of the 'best practice' technical solutions in other major electronic textual and archival projects as well as his training in library science and programming make him an ideal contributor and evaluator of project solutions.

Other Volunteer Programming consultants

- **Sindre Sorensen,** the subject of his current research is on base-text protocols, storage and and indexing systems for stand-off markup. He was a programmer with the Wittgenstein project, Bergen, Norway.

- **Michele Barbera,** a chief programmer for the HYPER, Discovery, and Talia projects, all supported by funding from the EU, he was a participant in the CaTT workshop at DMU in 2007. He works for Net7, Pisa, Italy. His role is to advise on technical interoperability between HRIT and other currently developing textual projects in Europe.

Graduate Research Assistants Kevin Caliendo, Richard Obenauf, Elizabeth Hanson, and Mark Owen--graduate students at Loyola University assigned to Peter Shillingsburg and involved in various ways in the project for periods of one semester.

Student Assistant: Joe Kaylor, CS student working with George Thiruvathukal

3. Goals and Accomplishments

a. History of Goal Development in the grant period

The goals for this project, set out at the beginning of the "Start Up Program" were to

- to identify best practices and define principles for on-line electronic infrastructure to support scholarly collaboration on primary literary and historical texts and for tools to enhance humanities scholarship,
- to build an open-source, collaborative, robust environment, in which to aggregate, link or cross-reference, edit, and share vetted primary documentary texts--along with their scholarly enhancements, analyses, and commentaries, in the form of markup, annotation, keyword tagging, linking, etc.
- to create a Collaborative Tagging Tool (CaTT) that responds to said principles and infrastructure, enabling ordinary humanist scholars to create sophisticated scholarly electronic editions and archives in a collaborative environment.

In June 2009, even before the Start Up grant officially began in September, we developed a much more ambitious and detailed version of HRIT, which we submitted to NEH in June 2009--not because we expected funding but because the exercise gave a great boost to the Start Up project. We followed that with an even more detailed funding request to NSF in October 2009, by this time with fingers crossed, but again primarily for the mental discipline of organizing and planning the details of HRIT. It was through these proposals and the weekly meetings of the HRIT team, that more and more of the details (modules and components) of the HRIT system were identified, leading to the full-scale project described in this report.

Design of the overall HRIT system and functions has been the primary work of Peter Shillingsburg who wrote the Development Handbook, the Glossary, the Tools description, and the Functions analysis.

Overall techincal guidelines, design, and protocols for environment, tool development and project record keeping has been the responsibility of George Thiruvathukal who wrote the HRIT Development Guidelines.

Programming has advanced on three general fronts:

First, development of a "services environment" (REST services) to link the tools and content projects in one environment, a provisional API (application programming interface) anticipating the interface functions for users, and a CMS (content management system) for developers and future collaborators;

Second, development of prototype tools (E-Hinman, Transparent, Magnify, ZOOM, Grids and Maps, editor (i.e. CaTT), Thoth, stripper, splitter, and the MVD/CorTex). Details about this work will be given below; and

Third, development of minimal (valence) markup. The chief programming duties for RESTful services fell to Pedro Prieto Alarcon; development of image manipulation tools, tagging tools, and a CMS (Mojulem) specifically designed for HRIT projects fell to Nicholas Hayward; and development of the MVD/CorTex tools fell to volunteer team member Desmond Schmidt, who had independently developed (as MVD and NMerge) programs that fulfilled the central functions of CorTex development and who modified them for HRIT use; Schmidt also developed the first API for HRIT, which was implemented in the RESTful services by Pedro Prieto Alarcon. Development of a Valence Markup system to compact and simplify Text Data Storage in the CorTex is the responsibility of George Thiruvathukal and his assistant, Joe Kaylor.

b. Current State of HRIT Project Development and Vision

The entry point to HRIT development and the public face of HRIT provides easy access to all HRIT documents, code depositories, and a slide show of screenshots. The primary development documents are:

Handbook provides the overall plan and rationales for all parts of HRIT;
Guidelines provides technical requirement for tools and services development;
Glossary provides definitions of terms used in other HRIT documents;
Functions describes the actions and manipulations required by the HRIT system;
Tools describes tools to be offered to users for developing and using HRIT.

The HRIT home page also gives access to demos, in-house communication discussions, galleries of project screenshots, HRIT associated projects, people, and links to related projects. We also have a full repository and history of code development with links to all HRIT web sites, and in-house records, fulfilling our commitment to open source, open access principles. The direct links to the repository for tools and demos are http://hrit.etl.luc.edu and http://hritwiki.ctsdh.luc.edu/demos

The vision for this project is to produce an electronic, online, open source, published site for editorial, analytical, and critical work, capable of launching and maintaining open-ended textual projects collaboratively developed by overlapping communities of scholars over time. What follows here is a brief summary. The full account is in the documents linked above to this report.

The projects we wish to enable, promote and even host are scholarly editions/archives/learning sites, called "Knowledge Sites" in Shillingsburg's *From Gutenberg to Google.* The central interest of humanities scholars is in the historical documents--the primary materials in manuscript and print--of literature because they form the foundation upon which the superstructure of analysis and new knowledge is built. The HRIT system will support the preservation and access to primary materials in an environment open to collaborative analysis, annotation, introduction, commentary and linking by scholars and students whose approaches and areas of expertise are wide ranging.

The underpinning principles of HRIT are that, in electronic representation, primary materials consist of both the bibliographical material (iconic autographic elements stored in pixel form) and the lexical texts (verbal allographic elements stored in Unicode form) and that together these aspects of primary textual materials form the basis for scholarly analysis, commentary and criticism (encoded and stored stand-off mode). These three aspects (material, lexical, and analytical) are developed and stored separately for reasons both technical and theoretical.

The most contentious aspect of this design is the separation of text from analytical markup, which currently is normally embedded in the text. We separate them, treating lexical data about textual variation and development as a *finite, controllable data set* whose value depends on its authenticity and accuracy, and treating analytical markup as *infinitely extensible add-on value* that comes from multiple sources and collaborators, each of whom will want to be identified and credited (perhaps for tenure and promotion purposes) for the contribution made to the published "Knowledge Site". A further advantage to storing analytical commentary as stand-off markup is that it gives users options when choosing the types of analysis they wish to see relative to the specific texts they wish to examine. Residual benefits of the HRIT system include the fact that when a scholar provides markup for one text of a work, that markup migrates to all other texts in the CorTex that are, at the point of markup, lexically congruent to the version of the work that is actually marked up. The result is that if a user requests from the data base a text that has not itself been marked up analytically, then markup already applied to lexically congruent texts at any given point in the text automatically is available with the new text. This is a result of the "merge and purge" compaction of text data in the CorTex and the stand-off nature of markup. The component structure and storage of HRIT tools and projects means that a content project can be launched when only some of its components have been completed, even if other parts are still in development.

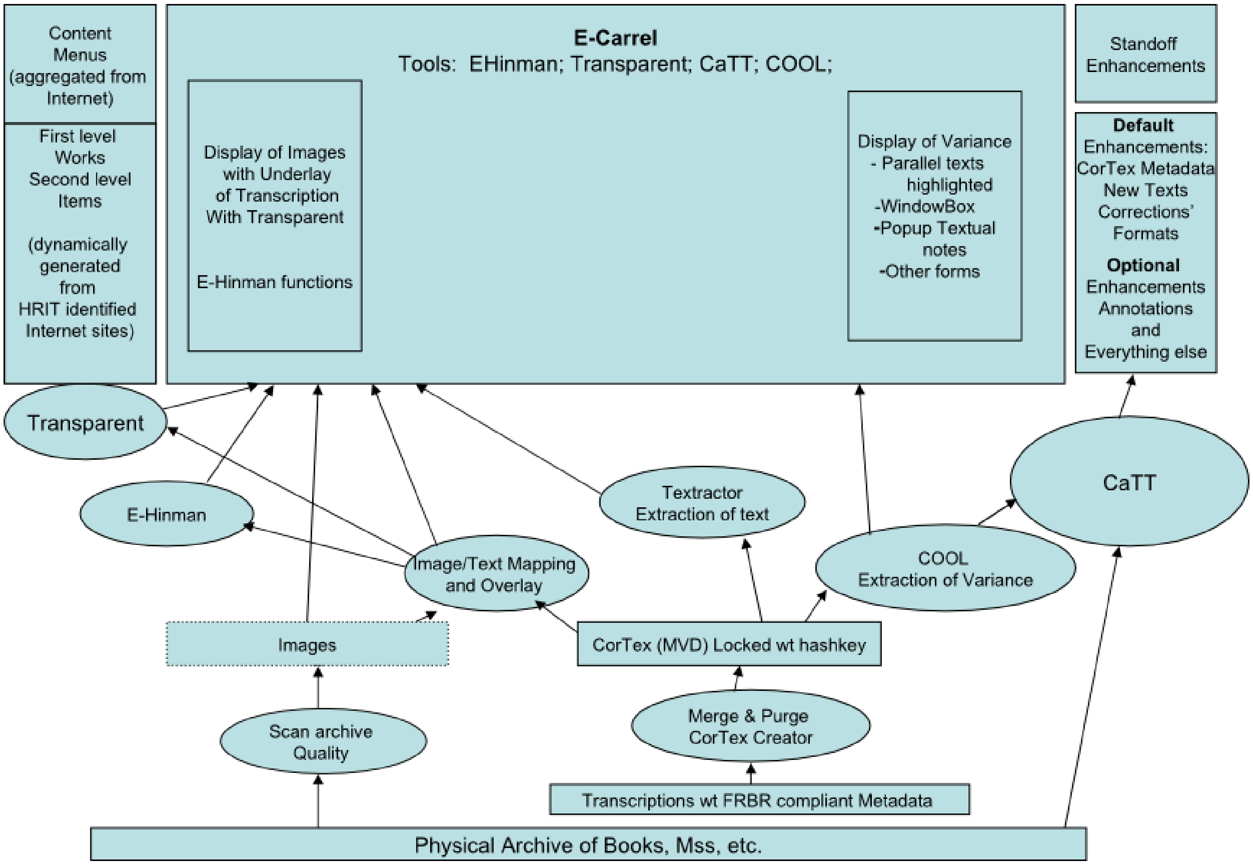
Content Menus (aggregated from Internet)
First level Works Second level Items (dynamically generated from HRIT identified Internet sites)
E-Carrel
Tools: EHinman; Transparent; CaTT; COOL;
Display of Images with Underlay of Transcription With Transparent
E-Hinman functions
Display of Variance - Parallel texts highlighted -WindowBox -Popup Textual notes -Other forms
Standoff Enhancements
Default Enhancements: CorTex Metadata New Texts Corrections' Formats
Optional Enhancements Annotations and Everything else
Transparent
E-Hinman
Textractor Extraction of text
CaTT
Image/Text Mapping and Overlay
COOL Extraction of Variance
Images
CorTex (MVD) Locked wt hashkey
Scan archive Quality
Merge & Purge CorTex Creator
Transcriptions wt FRBR compliant Metadata
Physical Archive of Books, Mss, etc.

Fig. 2 This schematic of the HRIT system reads from the bottom up--from the material archive up through tools and storage to the aggregation and display in the E-Carrel.

c. Tools and Frameworks

CMS for HRIT: A new CMS, for HRIT content projects, has been developed in modular form. A core publication framework is supplemented by plugins, such as image viewers or editors, which can be selected and added as project specific requirements dictate. It is also capable of hosting multiple projects within one installed framework, thereby enabling cross project research and the option to aggregate specified data. See mojulem for an initial example of a focused digital humanities framework to provide a modular set of tools. This is an early working example of phase two of the HRIT related Woolf Online project.

API: The outline or framework has been described in detail (in the Handbook); a RESTful HRIT framework has been constructed; an API has been proposed in two forms edu.luc.hrit and hritapirest; and component tools have been prototyped and placed in the framework.

Tagging tool: the original vision of one tool has evolved into several different functions (each with potentially different users, though anyone can use all three).

- THOTH (alpha version) is designed for users of TEI--it provides a learning and reference environment for TEI with quick access to the TEI Guidelines. It can also be referenced as required by applications performing queries such as TEI modules, elements, attributes, and relationships.
- The CaTT editor (alpha version) has now migrated to become a plugin, 'sema-tawy', within the 'mojulem' CMS framework providing editor functionality. It provides simple node creation and editing, including TEI, XML and text, plus collaborative functionality for members of a mojulem project. This will also form a working aspect of the latest phase of the Woolf Online project.
- A third tagging function, to be incorporated into CaTT, enables tagging existing CorTex content in stand-off mode as CorCode.
- CorCode and Standoff Markup
    - TEI Parser-Full and TEI Parser-Simple are programs for separating and supplying markup in standoff mode. They also provide import and rendering functionality for legacy TEI texts, supporting standard output in HTML with CSS styling. The parsers are currently designed for TEI, however they can easily be modified for use with other XML schemas. This allows us to broaden project support, both new and legacy.
    - formatter/splitter/stripper are programs for separating markup from the underlying text, allowing the separation of markup into "minimal encoding" and "heavy markup" as explained in the HRIT Handbook.
    - Both the Parsers and the splitter/stripper programs will become plugins for the CMS framework providing a facility to import or export TEI encoded texts in both legacy and new projects. They represent different approaches to the problem and demonstrate HRIT's ability to host optional tools for content manipulation and management.

MVD/CorTex: Multi-Version Documents and the nmerge tool is adapted to merge variant versions of a work into a MVD/CorTex, data from all the text files representing a work, purged of redundancy (caused by convergent texts). The MVD format and nmerge program are being gradually adapted to the HRIT environment. See further discussion follow the links to "MVD or CorTex" at MVD/CorTex.

Valence Markup: A conversion to dissolved or valence markup reduces the non-lexical aspects of a transcription file to a minimal form, preserving the following information at character level: semantically significant (e.g., italics), semantically significant white space (e.g. some forms of indentation); and mapping coordinates (to facilitate superimposition of text image and text transcription). Note: all other encoding, whether stripped from a legacy file or added after the project is launched is saved in stand-off mode.

Online infrastructure: HRIT was conceived as an online infrastructure in terms of an architecture, an abstract system for relating various text- and image-based components to tools for manipulation and study. That architecture is now understood and fleshed out in prototype form at hrit.etl.luc.edu, a page for demos, code repositories, and announcements. The goal now is to develop HRIT tools as online functions, as command line functions, and where possible, in mobile device form.

Stand-off markup is an essential element of HRIT's solution to collaborative work. We have built tools to strip markup from TEI-encoded texts and to convert that into a set of overlapping properties, which can be directly converted into a web-page. We have also built a basic demonstration of CorCode functionality.

Tools for images in beta versions include

- E-Hinman (which enables fade from the image of a page from one copy to another and the rapid "blinking" replacement of one image for its counterpart, like the mechanical Hinman Collator),
- Transparent (which enables a variety of views of an image in relation to its transcription, superimposed or juxtaposed).
- Grids and Maps Switcher and Parallel (which align images of texts with their transcriptions and enables location for image tags). These are also concept tools for use with an editor to allow a user/editor to map particularly difficult images to created transcriptions. ie: where OCR provides wholly unsatisfactory and unusable results. In effect, the editor can also become the mapper with little to no extra effort on their part.
- Magnify (self-explanatory)
- Zoom (ditto)
- Further experimental image tests and concepts can be found in the demo section of the HRIT Wiki

HRIT-lite, a first stage because the true goals of HRIT are very ambitious, is a version of HRIT that students of textual scholarship could use to create class projects. Thoth for tagging, the MVD/CorTex, and the image tools form the core of HRIT-lite.

A complete list of demos, concepts and experimental tools can be found on the HRIT Wiki demo page.

4. Platform

HRIT is a *polyglot* environment using many languages: PHP, Java, C/C++, Python, etc.

- HRIT employs modern software engineering practices to ensure that tools and (programming) libraries can be comprised in multiple ways: as web applications, web services, and standalone tools (the old-world command line). This flexibility is achieved by having proper "binding surfaces" such as *language interfaces* and/or *web services* (e.g. REST).
- HRIT emphasizes free/open source in everything, which is in keeping with the desire to disseminate widely.
- HRIT encourages others to use all--or part--of the developed solutions wherever appropriate--even in a competing setting. Its usefulness is defined not only by whether HRIT tools are used as offered, but also by whether HRIT code ends up being useful for *someone else's* tools.

5. Outcomes

"Graduation" from the NEH Startup phase of HRIT development is truly more of a "commencement" than having reached a final goal. The design of HRIT's environment, tools, frameworks, interactions, and capabilities has reached a well-developed stage, as is shown in the Developer's Manual, Technical Guidelines, and Code Repositories. Many of the building blocks for HRIT are in alpha or beta form, but it is not yet the case that an HRIT project can be initiated and brought to display stage.

6. Lessons Learned

First and foremost, we were confirmed in our initial conviction that HRIT is not a Humanities project with support from Computer Science. It is instead a dual-edged research and development project requiring innovation and development in both disciplines. The joint leadership of Shillingsburg (humanist) and Thirvathukal (Computer Scientist) was essential to the organization of the work methods and the innovations in technical solutions. Furthermore, it has been crucial to our success, such as it is, that Nick Hayward has a PhD in a humanist discipline (Ancient History and Archaeology) and an MSc in Computer Science, and that our surprise partner, Desmond Schmidt, has two PhDs, one in Classics and one in Information Technology. The push and pull that has come from cross-discipline collaboration has improved the overall design of HRIT, its methods for developing architectures, its approach to open source programming, and it methods of record keeping. These things we imagined to be so when we began; they have proven to be so in practice. The support given by Steven Jones because of his interest in both textual studies and gaming, has also led to innovations and to the push to design HRIT for both high-end computing and mobile devices. We are not there yet, but the goals for HRIT development have been significantly influenced by the fact that we are a multi-disciplinary team of professional, not a one-person project with a team of assistants.

Second, we learned that long-distance team work is difficult and bumpy, not an easy given, based on the availability of conference calls. Face to face meetings seem to be required in order to develop trust and to overcome the communication gaps that spring up in any complicated collaboration. Break downs in communication could have been detrimental to the project, but instead they have led to new solutions and redoubled efforts. Face to face meetings will continue to be necessary, but the communication system and the shared code repositories have built a remarkable amount of trust in the HRIT team. Team meetings have been synchronised with conferences; we have met in Chicago, Brussels, College Park Maryland, and London, often with extra-project financial support for travel.

Third, we learned that it was not necessary to make programmers work in lock step in one language and one platform. The versatility of HRIT is owing in part to attempts to accommodate what each programmer wanted to do. This was made possible by the modular structure of HRIT's parts and the REST environment where all HRIT solutions reside and interact. HRIT programmers work with Mac, Windows, and Linux operating systems; they use PHP, Java, C/C++, Python, Apache, etc.; they work in England, Spain, Australia and the US.

7. Future Plans

Loyola University Research Office has give the HRIT project an $8K bridge to continue working after the expiration of the NEH Startup grant. We have applied to NEH Preservation and Access program for three years of additional funding to implement and complete the tools and environment and develop HRIT projects. We already have funding for the Woolf Online project. Like most content-oriented funding of humanities projects, the Woolf Online project has a deadline and a mountain of work on Woolf. It is incorporating as much of the HRIT design as it can, but Woolf Online will have to be finished before we can hope to have all of HRIT in place.

Likewise, we have applied to NEH to fund an electronic archive of Thomas Hardy's fiction, bringing together a team of eight Hardy scholars to do the "content work" for an archive designed to HRIT specifications.

We are also applying to the NSF for funding the implementations of HRIT to mobile devices and to develop HRIT tools for students. We know we will not get all the funding we have applied for, but without further external funding the aims of HRIT will be accomplished only at a very slow pace.

8. Publications and Presentations

The HRIT Day Conference was held on Sat. October 30,2010, with Laura Mandell, Desmond Schmidt, George Thiruvathukal and Peter Shillingsburg presenting.

Steven E. Jones, Peter Shillingsburg, George K. Thiruvathukal.

- E-Carrel: An Environment for Collaborative Textual Scholarship. *Proceedings of the Chicago Colloquium on Digital Humanities and Computer Science*, Vol 1, No 2 (2010)

Nicholas Hayward

-'e-Carrel: Looking to the future with one foot in the past', EIRI/CCH Conference on the Digitisation in the Humanities, Keio University, Tokyo, Japan - March 2010

Desmond Schmidt and Domenico Fiormonte

- Documenti multiversione: una soluzione per gli artefatti testuali del patrimonio culturale/ Multi-version documents: a digitisation solution for textual cultural heritage artefacts. Intelligenza artificiale, IV.1, Dec 2010, 56-61.

Desmond Schmidt

- Merging Multi-Version Texts: a General Solution to the Overlap Problem, in Balisage: The Markup Conference Proceedings 2009, Montreal, August.

Desmond Schmidt and Robert Colomb

- A Data Structure for Representing Multi-version Texts Online. International Journal of Human Computer Studies 67.6 (June, 2009), 497-514.

Peter Shillingsburg

- *From Gutenberg to Google: Electronic Representations of Literary Texts* (CUP 2006) Not a presentation, but from my point of view a basic document underpinning HRIT
- "HRIT and Collation" COST Workshop, Brussels, Nov. 20, 2009

-“Lessons in Computing the Humanities” Book Logic conference Univ. of Sydney, Australia, Feb. 2010

- "What Google Owes to Gutenberg" Textual Studies conference Saitama University, Tokyo, Feb. 2010
- “HRIT for electronic scholarly editing” day conference on electronic editions, Loyola, Oct. 2010
- Roundtable discussion, Modernist Studies Assoc., Vancouver, Nov. 11-12, 2010 (HRIT related)
- Public Lecture, Antwerp City Library, “From Gutenberg to Google”, Nov. 24, 2010
- “Textual Worth and Textual Tools” ESTS, Pisa Italy, Nov. 25-27, 2010.

9. HRIT Application to Content Projects

- Woolf Online, funded by NEH
- Thomas Hardy Archive, funding pending
- Harpur Poetry, Australia, funded allied project

The first full scale application of HRIT principles and tools to a content project by members of the HRIT team is the Woolf Online Project, funded by NEH, and built on a pilot project initiated in the very early stages of HRIT development.

A significant desideratum in applying HRIT to a content project was a content management system. Nicholas Hayward created a new CMS, not because CMSs were unavailable but because they were not designed for specific content use or research within humanities, and in particular did not meet the demands of HRIT projects. Existing CMS frameworks could have been retooled and repurposed for use with HRIT services, such as Joomla or Drupal, but these CMSs are not designed from the ground up for use with TEI, MVD etc. Although not wholly unsuitable for the job, experience using Drupal, for example, indicated what was really necessary to get a project created, and how best to target this for digital humanities. Image management, taxonomy structure, XML parsing, and editing are all aspects of a digital project that are quite different from mainstream CMSs.

The HRIT CMS is a content management framework for the development of digital humanities projects and archives. It features a core base structure, which includes user login, registration and management, and basic textual node creation. It also includes a full admin structure to maintain and manage the site's structure, properties, content etc from within a browser. Then, dependent upon a project's set of requirements, plugins can be added to manage content and project requirements. The site's content and structure is tightly managed by a taxonomy structure that is controlled by the project, and in turn affects the content's management, display and relation to each other. A flexible plugin architecture enables projects to use more than one particular type of textual record, and to use TEI or another XML schema, plain text, html etc. Other plugins can manage editing, parsing, image display and manipulation, research, and so on. The number of plugins will grow as the framework is used for different projects, which in turn will allow for more options for other, future projects.